

**Тема:** Робота із сценою та шарами в Cocos2d-x v. 3.x

**Мета:** Дізнатись що таке сцена та шари. Навчитись переходити між сценами та налаштовувати анімацію переходу. На прикладі описаного багатосценного проекту дізнатись особливості зв'язування різних сцен проект в один цілісний проект.

### **Зміст:**

1. Створення сцени.
2. Перехід між сценами
3. Створення структури гри.

Сцена - це контейнер, який містить у собі Sprite, Labels, Nodes та інші об'єкти гри. Сцена відповідає за логіку гри та інтерфейс, тобто те що відображається на екрані гаджету. Щоб почати гру, потрібно мати принаймні одну сцену. Ви можете подумати про це, як про фільм. The

Сцена - це те, що працює, і користувачі бачать, що відбувається в режимі реального часу. Можна мати будь-яку кількість сцен-об'єктів у грі і здійснювати переходи між сценами. Cocos2d-x забезпечує анімацію переходу, і навіть можна перейти зі сцени на сцену з ефективними ефектами.

## **Створення сцени**

Створити Сцену дуже легко:

```
auto scene = Scene::create();
```

Для позиціонування спрайтів на сцені можна використовувати поточні розміри сцени.

Розміри екрану можна отримати використавши метод

```
Director::getInstance(). Для цього потрібно прописати:  
Size size = Director::getInstance()->getWinSize();
```

Клас **Size** має два методи width і height, які визначають граничні значення координат екрану.

Ці методи можна використовувати, скажімо для позиціонування вузлів.

```
mySprite1->setPosition(Vec2(size.width / 2, size.height / 2));
```

в кодї вище, спрайт займе саме центр екрану.

### Створення елементарної сцени.

Для правильної побудови сцени, та розташування на ній елементів слід пам'ятати що система координат починається з нижнього лівого кута. Саме там знаходиться початок координат (0, 0). І саме там по замовчуванню з'являтимуться всі елементи.

```
auto dirs = Director::getInstance();
Size visibleSize = dirs->getWinSize();
auto myScene = Scene::create();

auto label1 = Label::createWithTTF("My Game", "fonts/Marker Felt.ttf", 36);
//label1->setPosition
label1->setPosition(Vec2(visibleSize.width / 2, visibleSize.height / 2));
myScene->addChild(label1);

auto sprite1 = Sprite::create("1.png");
sprite1->setPosition(Vec2(200,200));
myScene->addChild(sprite1);
```

Початкові налаштування екрану (встановлення фону, BackGround спрайтів) можна задавати в функції `bool HelloWorld::init()`.

Наприклад:

```
bool HelloWorld::init()
{
    if (!Layer::init())
    {
        return false;
    }
    Size size = Director::getInstance()->getWinSize(); // отримання розміру екрану
    auto sprite = Sprite::create("res/run_01.png");
    sprite->setPosition(Vec2(size.width / 2, size.height / 2));
    this->addChild(sprite);
    return true;
}
```

### Перехід між сценами

Без переміщення між сценами неможливе, Cocos2d-x володіє великим функціоналом та способами переміщення з однієї сцени в іншу.

Cocos2d-x використовує стек для керування сценами, що є системою "Last-In-First Out" (LIFO), що запускає останню сцену в стеці. Нижче наводяться основні методи використовується для переміщення між сценами:

- **Запис сцени (Pushing a scene):** цей метод прописує певну сцену в стек, зберігаючи при цьому поточну сцену, і призупиняє її виконання. Приклад для реального світу для це відбувається, коли ви натискаєте кнопку паузи, пауза сцена буде натиснута на стека, а сцена гри все-таки існує.

- **Відтворення сцени (Popping a scene):** цей метод видаляє верхню / поточну сцену зі стеку. Реальним прикладом для цього є відновлення призупиненої сцени, поточного сцена (сцена - пауза) видаляється і повертається до сцені гри.

- **Заміна сцени (Replacing a scene):** цей метод замінює поточну сцену на нову сцену, по суті спливаюча поточна сцена, а потім натискання нової сцени на стек. Прикладом реального є те, коли гравець помирає і йде нова гра на сцені.

Очевидно, що потрібно буде переміщення між сценами у проекті гри.

Маємо сцену:

```
auto myScene = Scene::create();
```

**runWithScene ()** – метод який використовується лише для першої сцени. Це спосіб почати гру з конкретної сцени.

```
Director::getInstance()->runWithScene(myScene);
```

**replaceScene ()** - повністю замінює сцену.

```
Director::getInstance()-> replaceScene (myScene);
```

**pushScene ()** – Задіявання нової сцени із збереженням попередньої сцени

```
Director::getInstance()-> pushScene (myScene);
```

**popScene ()** – Задіявання нової сцени із стиранням попередньо задіяної сцени.

```
Director::getInstance()-> popScene (myScene);
```

## Ефекти переходу між сценами.

Можете додавати візуальні ефекти переходів між сценами:

```
auto myScene = Scene::create();
```

```
    // Ефект Fade
```

```
    Director::getInstance()->replaceScene(TransitionFade::create(0.5, myScene,  
Color3B(0, 255, 255)));
```

```
    // FlipX
```

```
    Director::getInstance()->replaceScene(TransitionFlipX::create(2, myScene));
```

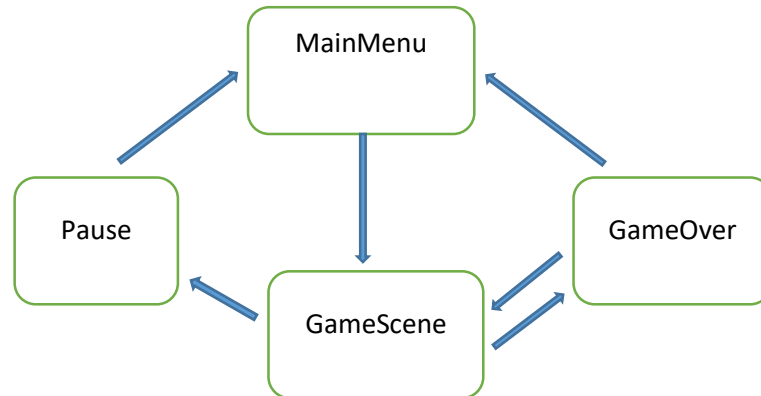
```
    // Ефект Slide In
```

```
    Director::getInstance()->replaceScene(TransitionSlideInT::create(1, myScene));
```

## Створення структури гри.

Очевидно, що найголовнішим в грі це добре продумана структура сцен, організована дієва навігація по сценах.

Для цього створимо 4 сцени та пропишемо навігацію між ними. Кожна сцена передбачатиме свою пару \*.cpp та \*.h файла.



мал. 1

Розробимо структуру гри яка передбачатиме 4 сцени, а також зв'язки між ними. Схематично структура зображена на схемі вище.

Тепер створимо нові файли для наших сцен і добавимо їх в наш проект. Кожна сцена матиме свій клас, який буде викликатись з інших сцен. Найпростіше це продублювати уже наявні в згенерованому проекті файли HelloWorld.

Створимо повністю нове середовище для проекту, замінивши пропоновані назви на власні.

Для початку створимо новий проект, і перед першим запуском зробимо зміни.

1. Заміняємо HelloWorldScene.h на MainMenu.h
2. Відкриваємо файл і робимо зміни в ньому:

a. заміняємо

```
#ifndef __HELLOWORLD_SCENE_H__
```

на

```
#ifndef __MAINMENU_H__
```

b. заміняємо

```
#define __HELLOWORLD_SCENE_H__
```

на

```
#define __MAINMENU_H__
```

c. Змінюємо назву класу

```
class HelloWorld : public cocos2d::Scene
```

на

```
class MainMenu : public cocos2d::Scene
```

d. Видозмінюємо функцію **CREATE\_FUNC(MainMenu);**

е. Видаляємо функцію

```
void menuCloseCallback(cocos2d::Ref* pSender);
```

3. Заміняємо

HelloWorldScene.cpp

на

MainMenu.cpp

4. Відкриваємо файл MainMenu.cpp і робимо зміни в ньому:

а. **#include "MainMenu.h"** замість **#include "HelloWorldScene.h"**

б. Змінюємо і інший код, до вигляду:

```
USING_NS_CC;  
Scene* MainMenu::createScene()  
{  
    auto scene = Scene::create();  
    auto layer = MainMenu::create();  
    scene->addChild(layer);  
  
    return scene;  
}  
bool MainMenu::init()  
{  
    if (!Layer::init())  
    {  
        return false;  
    }  
    return true;  
}
```

5. Наступним кроком змінюємо файл ppDelegate.cpp

а. відкриваємо файл ppDelegate.cpp

б. Заміняємо заголовочний файл HelloWorld на **#include "MainMenu.h"**

с. Заміняємо функцію

```
auto scene = HelloWorld::createScene();
```

на

```
auto scene = MainMenu::createScene();
```

**ЗАУВАЖЕННЯ!** Для запуску проекту під андроїд треба буде змінити Android.mk файл в папці GAMENAME\proj.android-studio\app\jni\

```
LOCAL_SRC_FILES := hellocpp/main.cpp \  
                    ../../../../Classes/AppDelegate.cpp \  
                    ../../../../Classes/MainMenu.cpp
```

**ЗАУВАЖЕННЯ!** Для запуску проекту під Windows необхідні наступні додаткові зміни в файлах папки proj.win32:

файл NewGame.vcxproj -

```
<ClCompile Include="..\Classes\HelloWorldScene.cpp" />  
    <ClInclude Include="..\Classes\HelloWorldScene.h" />
```

файл NewGame.vcxproj.filters -

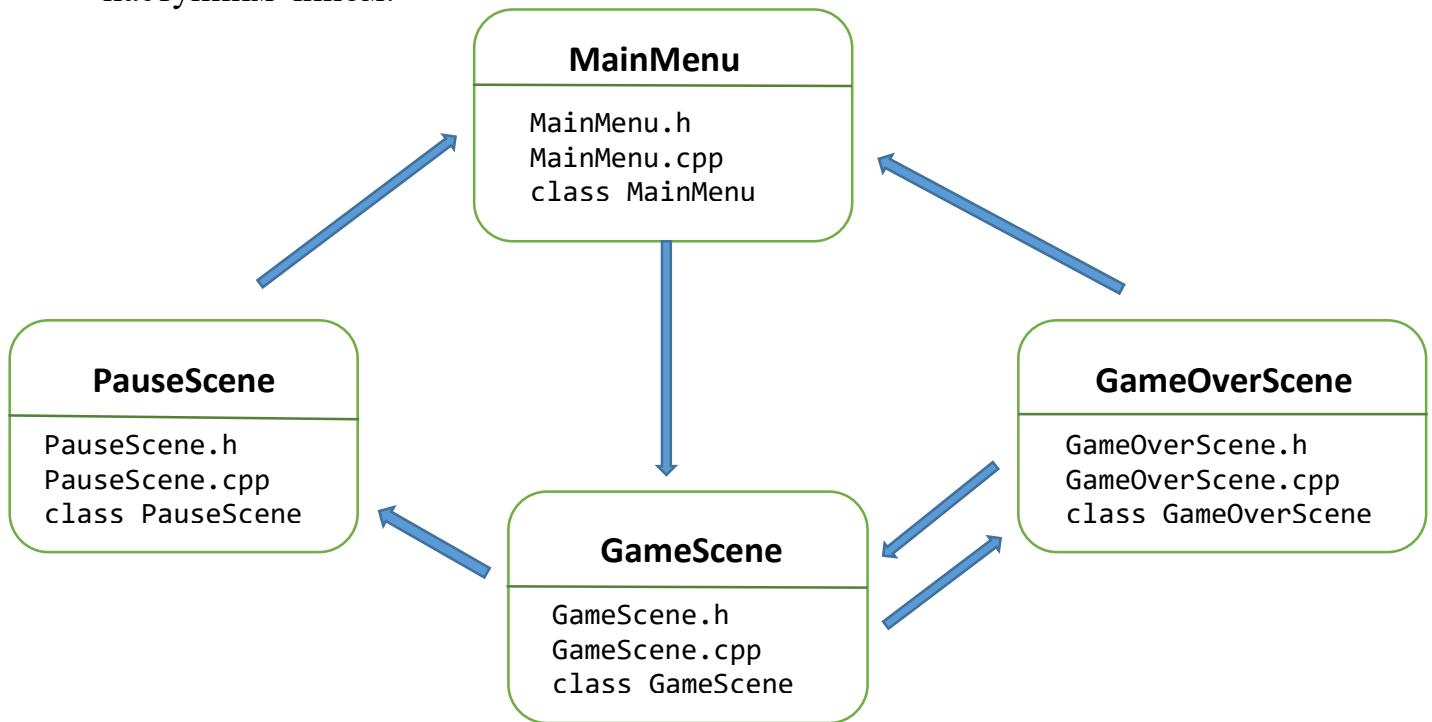
```
<ClCompile Include="..\Classes\HelloWorldScene.cpp">
```

```
<CInclude  
Include="../../Classes/HelloWorldScene.h">
```

Після проведених змін варто запустити та скомпілювати проект.

Тепер перейдемо до створення файлів \*.cpp та \*.h для сцен: паузи, головної сцени гри та сцени закінчення гри.

Загальна схема проекту з усіма файлами та класами виглядатиме наступним чином:



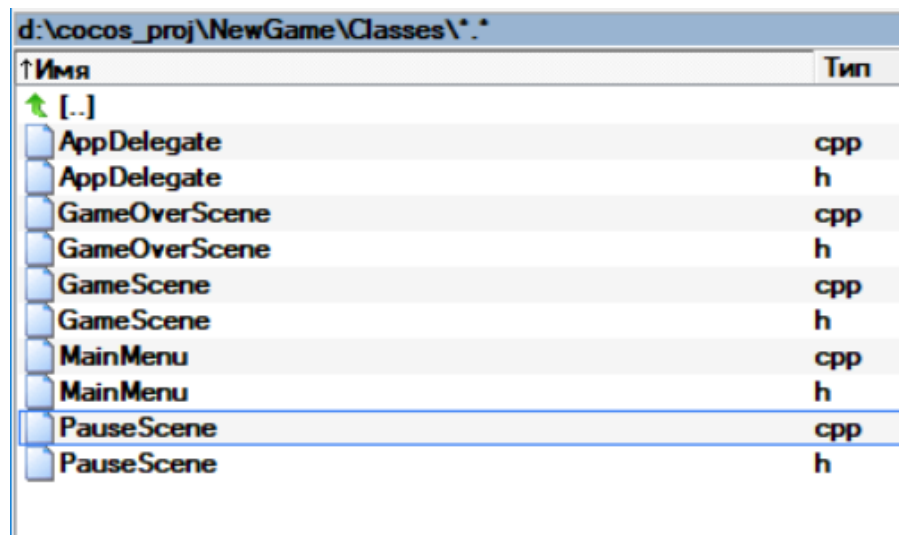
мал. 2

Створимо файли для головної сцени гри.

1. Продублюємо наявні файли **MainMenu.cpp** та **MainMenu.h** і поміняєм назви на **GameScene.cpp** та **GameScene.h**
2. Далі потрібно здійснити всі необхідні заміни в цих файлах, подібні до тих, які робились коли редагувався **HelloWorld** на **MainMenu**. Відповідний клас назовемо **GameScene**.

За такою ж послідовністю створюється ще дві сцени **PauseScene** та **GameOverScene**.

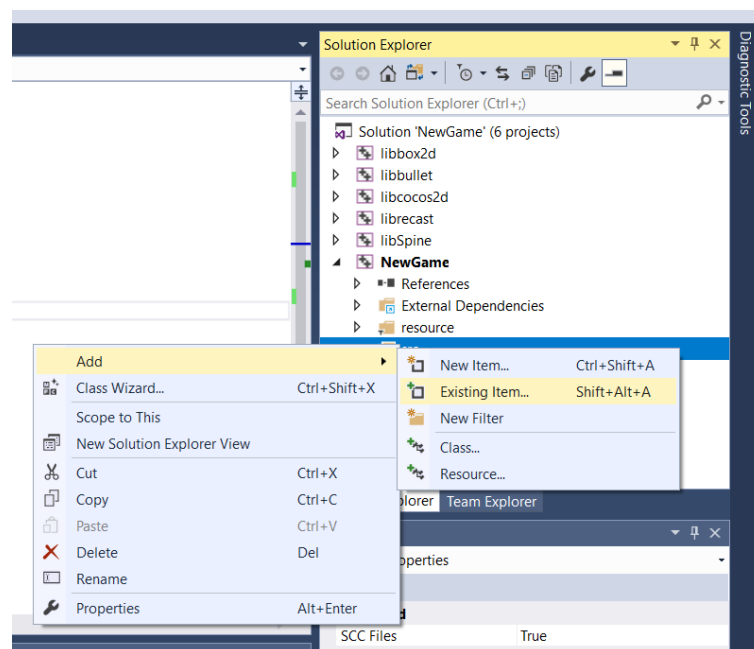
Структура папки **Classes** виглядатиме наступним чином:



Имя	Тип
[..]	
AppDelegate	cpp
AppDelegate	h
GameOverScene	cpp
GameOverScene	h
GameScene	cpp
GameScene	h
MainMenu	cpp
MainMenu	h
PauseScene	cpp
PauseScene	h

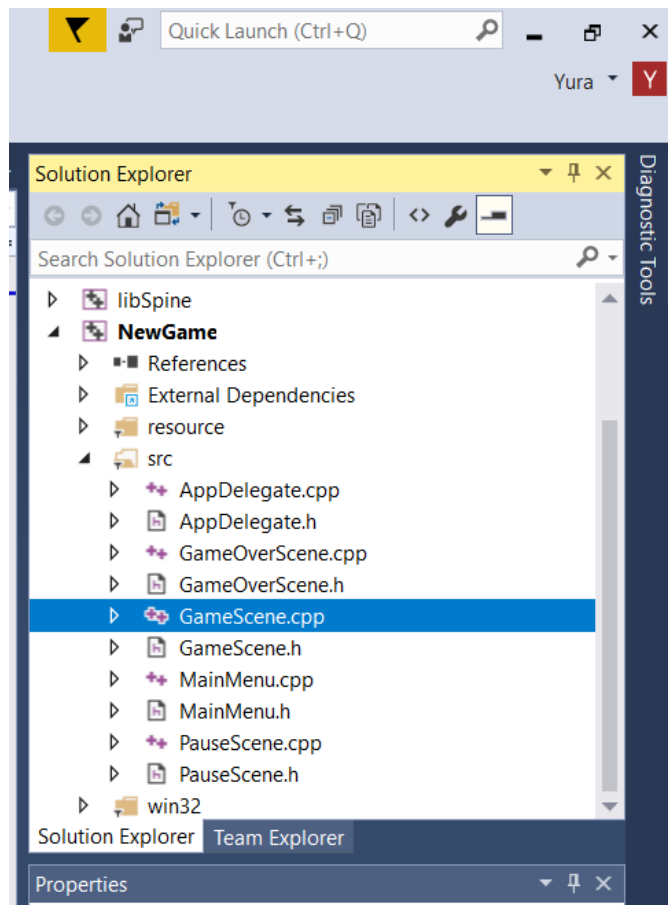
мал. 3

Наступним кроком слід додати створені файли в проект, для цього слід виконати команду **Existing Item ...** з меню **Add** контекстного меню папки **Src**:



мал. 4

Після виконаних дій матимемо наступну структуру проекту



мал. 5

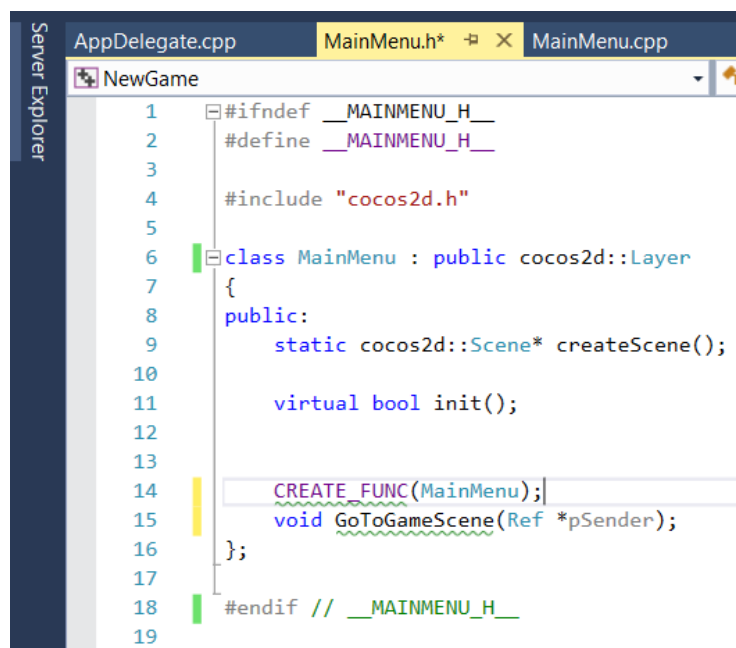
## Зв'язування сцен та створення функцій переходів між сценами.

1. В файлі `MainMenu.h` додаємо функцію

```
void GoToGameScene(Ref *pSender);
```

Це оголошення функції, яка буде викликана, коли гравець натискає на кнопку відтворення зі сценарію Головного меню, щоб замінити її сценою гри.





мал. 6

2. Наступний код додаємо в **MainMenu.cpp**

```
#include "GameScene.h"
```

```
... ..
```

```
void MainMenu::GoToGameScene(Ref *pSender)
{
    auto scene = GameScene::createScene();
    Director::getInstance()->replaceScene(scene);
}
```

У вищевказаному коді перший рядок використовується для включення головної сцени гри, щоб був доступ до сцени. Функція **GoToGameScene** спочатку створює локальну сцену на основі **GameScene**, а потім замінює поточну сцену, використовуючи Cocos2d-x режисер та функцію заміни сцени.

Загальний вигляд програми матиме наступний вигляд:

```
1  #include "MainMenu.h"
2  #include "GameScene.h"
3
4  USING_NS_CC;
5
6  Scene* MainMenu::createScene()
7  {
8      auto scene = Scene::create();
9      auto layer = MainMenu::create();
10     scene->addChild(layer);
11
12     return scene;
13 }
14
15 bool MainMenu::init()
16 {
17     if (!Layer::init())
18     {
19         return false;
20     }
21     return true;
22 }
23
24 void MainMenu::GoToGameScene(Ref *pSender)
25 {
26     auto scene = GameScene::createScene();
27     Director::getInstance()->replaceScene(scene);
28 }
29
```

мал. 7

3. Переходимо до налаштування файлу **GameScene.h**  
Тут дописуємо наступний код:

```
void GoToPauseScene(Ref *pSender);
void GoToGameOverScene(Ref *pSender);
```

4. Дописуємо код і до **GameScene.cpp**

```
#include "PauseScene.h"
#include "GameOverScene.h"
... ..
void GameScene::GoToPauseScene(cocos2d::Ref *pSender)
{
    auto scene = PauseScene::createScene();
    Director::getInstance()->pushScene(scene);
}
void GameScene::GoToGameOverScene(cocos2d::Ref *pSender)
{
    auto scene = GameOverScene::createScene();
    Director::getInstance()->replaceScene(scene);
}
```

5. Переходимо до налаштування файлу **GameOverScene.h**  
Тут дописуємо наступний код:

```
void GoToGameScene(Ref *pSender);
```

```
void GoToMainMenu(Ref *pSender);
```

## 6. Допишуємо код і до `GameOverScene.cpp`

```
#include "GameScene.h"
#include "MainMenu.h"
... ..
void GameOverScene::GoToGameScene(cocos2d::Ref *pSender)
{
    auto scene = GameScene::createScene();
    Director::getInstance()->replaceScene(scene);
}
void GameOverScene::GoToMainMenu(cocos2d::Ref *pSender)
{
    auto scene = MainMenu::createScene();
    Director::getInstance()->replaceScene(scene);
}
```

## 7. Переходимо до налаштування файлу `PauseScene.h` Тут дописуємо наступний код:

```
void Resume(Ref *pSender);
void GoToMainMenu(Ref *pSender);
void Retry(Ref *pSender);
```

У наведеному коді перша оголошена функція буде викликана коли користувач натискає кнопку повернення в гру. Друга оголошена функція буде викликана, коли гравець клацає на кнопці головного меню, в цьому випадку викликається сцена головного меню, замінюючи поточну сцену. Третя оголошена функція буде викликана, коли гравець натискає кнопку "Повторити спробу".

## 8. Допишуємо код і до `PauseScene.cpp`

```
#include "GameScene.h"
#include "MainMenu.h"
... ..
void PauseScene::Resume(cocos2d::Ref *pSender)
{
    Director::getInstance()->popScene();
}
void PauseScene::GoToMainMenu(cocos2d::Ref *pSender)
{
    auto scene = MainMenu::createScene();
    Director::getInstance()->popScene();
    Director::getInstance()->replaceScene(scene);
}
void PauseScene::Retry(cocos2d::Ref *pSender)
{
    auto scene = GameScene::createScene();
```

```
Director::getInstance()->popScene();  
Director::getInstance()->replaceScene(scene);  
}
```

### Виклик інших сцен.

Для виклику сцени використовується функція **CC\_CALLBACK\_1()**. В якості аргументів якої відображається відповідна функція **GoTo**.

Для прикладу розглянемо код, який розмістить на сцені **MainMenu** елемент **playItem**, нажавши на який викличеться **GameScene**. Даний код реалізовується в функції ініціалізації **MainMenu::init()** файлу **MainMenu.cpp**

```
bool MainMenu::init()  
{  
    if (!Layer::init())  
    {  
        return false;  
    }  
    auto playItem = MenuItemImage::create("1.png", "2.png",  
CC_CALLBACK_1(MainMenu::GoToGameScene, this));  
    auto menu = Menu::create(playItem, NULL);  
    menu->alignItemsVerticallyWithPadding(200);  
    this->addChild(menu);  
    return true;  
}
```

Функція **MenuItemImage::create** має три аргументи. Перші два це зображення самого пункту меню: коли він не активний (перший аргумент) і коли активний (другий аргумент). Третій аргумент вказує на дію, яку потрібно виконати при активації елемента меню. Як правило це функція зворотнього виклику, або лямбда-функція.

### Список літературних джерел по темі:

1. **Siddharth Shekar** Learning Cocos2d-x Game Development // 2014 Packt Publishing, p.266, ISBN 978-1-78398-826-6
2. **Raydelto Hernandez** Building Android Games with Cocos2d-x // 2015 Packt Publishing, p. 160, ISBN 978-1-78528-383-3
3. **Roger Engelbert** Cocos2d-x by Example Beginner's Guide Second Edition // 2015 Packt Publishing, p. 270, ISBN 978-1-78528-885-2
4. **Akihiro Matsuura** Cocos2d-x Cookbook // 2015 Packt Publishing, p. 255, ISBN 978-1-78328-475-7
5. **Karan Sequeira** Cocos2d-x Game Development Blueprints // 2015 Packt Publishing, p. 392, ISBN 978-1-78398-526-5
6. **Frahaan Hussain, Arutosh Gurung, Gareth Jones** Cocos2d-x Game Development Essentials // 2014 Packt Publishing, p. 392, ISBN 978-1-78398-786-3